CS 134: Networks
Prof. Yaron Singer Spring 2017

1 Overview

So far, we've primarily considered *social* networks, where we simulate relationships, the spread of trends/infections, competitive decisions between players, etc. In general, our graphs have generally provided opportunities for some kind of social or economic interaction. Now, we consider a different type of network, in which our nodes being connected are units of information, and the edges join related information nodes in some way. We refer to such a network as an **information network**. One prime example of an information network that we look at in this lecture is the World Wide Web. In this lecture, we begin with some context about the Web, and then look further back into the history of information networks that led up to the Web, and finally check out one powerful method of "node endorsement in such a network, namely **PageRank**.

2 The World Wide Web

At a fundamental level, the Web is an application developed to let people share information over the Internet; it was created by Tim Berners-Lee from 1989 - 1991. More specifically, it allows for a way to make documents easily available to anyone on the Internet, in the form of **Web pages** that users create and store on a publicly accessible part of their computer. In addition, it provides a way for others to easily access such Web pages via a **browser** that can connect to the public spaces on computers across the Internet and retrieve the Web pages stored there.

The critical design principle embedded in the Web - the decision to organize the information using a network metaphor, was first proposed by Vannevar Bush in his article "As We May Think" in the Atlantic in 1945. In particular, Bush observed that traditional methods for storing information in a book, a library, or a computer memory are highly linear they consist of a collection of items sorted in some sequential order. On the other hand, our conscious thinking relies on *associative memory*, where we develop insights based on connected relating things in our mind. Bush therefore called for the creation of information systems that mimicked this style of memory; he imagined a hypothetical prototype called the *Memex* that functioned very much like the Web, consisting of digitized versions of all human knowledge connected by associative links. In this way, Bush foreshadowed the modern Web.

The network aspect is what truly defines a web of Web pages such as the one in Figure 1. In writing a Web page, you can annotate any portion of the document with a virtual link to another Web page, allowing a reader to move directly from your page to this other one. The set of pages on the Web thereby becomes a graph, and in fact a directed graph: the nodes are the pages themselves, and the directed edges are the links that lead from one page to another. In fact, the use of a network structure truly brings forth the globalizing power of the Web, by allowing anyone authoring a Web page to highlight a relationship with any other existing page, anywhere in the world.

The decision to model the Web as a network is an application of a computer-assisted style of authoring known as **hypertext**, where the goal is to replace the traditional linear structure

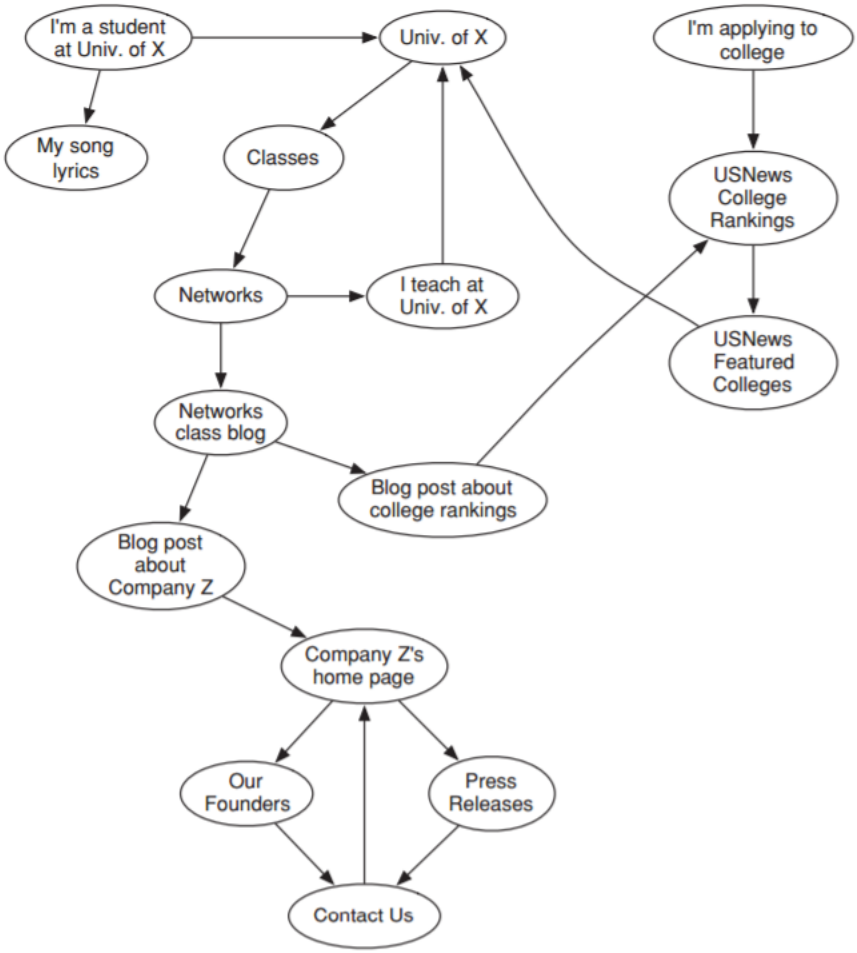


Figure 1: An example of a directed graph formed by the links among a small set of Web pages (*Kleinberg 13.3*).

of text with a network structure, in which any portion of the text can link directly to any other part. Some important intellectual precursors to hypertext include scholarly citations, where nodes represent papers and directed edge (x, y) represents paper x citing paper y, and encyclopedic cross-references, where related articles may reference each other, such that one might reach a very different topic by simply traversing a few edges in such a network. These short paths between seemingly distant concepts reflect an analogue, for information networks, of the "six degrees of separation" phenomenon we saw earlier in lecture.

In the early phase of the Web, most pages were relatively static documents, and most links served primarily **navigational** functions to transport you from one page to another according to the relational premise of hypertext. Now, on the other hand, the powerful computation available at the other end of a link is often brought more directly into play: **transactional** links such as

"Upload Image" or "Add to Cart" trigger complex programs on the computer hosting the page. While a lot of content on the Web now has a primarily transactional nature, this content still remains largely linked together by a navigational backbone it is reachable via relatively stable Web pages connected to each other by more traditional navigational links. This is the portion of the Web we will focus on in our analysis of its global structure. For instance, search engines try to assess whether the content they are collecting is relatively stable and intended for public consumption, and they tend to collect content that is reachable via navigational links.

In thinking about the Web as a network, we need to appreciate that its fundamentally *directed* nature makes it different from many of the networks we've considered thus far. In order to better understand directed graphs, we have the following definitions:

Definition. *A **path** from a node A to a node B in a directed graph is a sequence of nodes, beginning with A and ending with B, with the property that each consecutive pair of nodes in the sequence is connected by an edge pointing in the forward direction.*

Definition. *A directed graph is **strongly connected** if there is a path from every node to every other node.*

Definition. *A **strongly connected component** (SCC) in a directed graph is a subset of the nodes such that*

- *every node in the subset has a path to every other;*
- *the subset is not part of some larger set with the property that every node can reach every other*

Figure 2 shows the strongly connected components of the directed graph from Figure 1. In this case, we can see how the SCCs serve as a compact summary of how "reachable" a node is in a directed graph. Namely, if A and B belong to the same SCC, or if there is a forward path from the SCC of A to the SCC of B, then there is a path from A to B in the graph; if there is no way to do this, then there is no path from A to B.

3 PageRank

One of the greater challenges in the field of information network research is the identification of "prominent" pages, namely sites that either endorse (provide links to) many other sites or are themselves heavily endorsed. Oftentimes, these two types of pages are disjoint, and it may be difficult to find sites on related topics (e.g. competing businesses in the same field will generally not link to each other on their websites). One instinctive strategy to determine prominent pages is "voting using in-degree," where we collect a sample of pages relevant to a search query and determine the sites that are linked to the most by those pages. We will study this technique and other fundamental ideas behind page prominence on Wednesday, but for now we'll take a look at a powerful algorithm that measures the importance of pages based on linkage, namely PageRank.

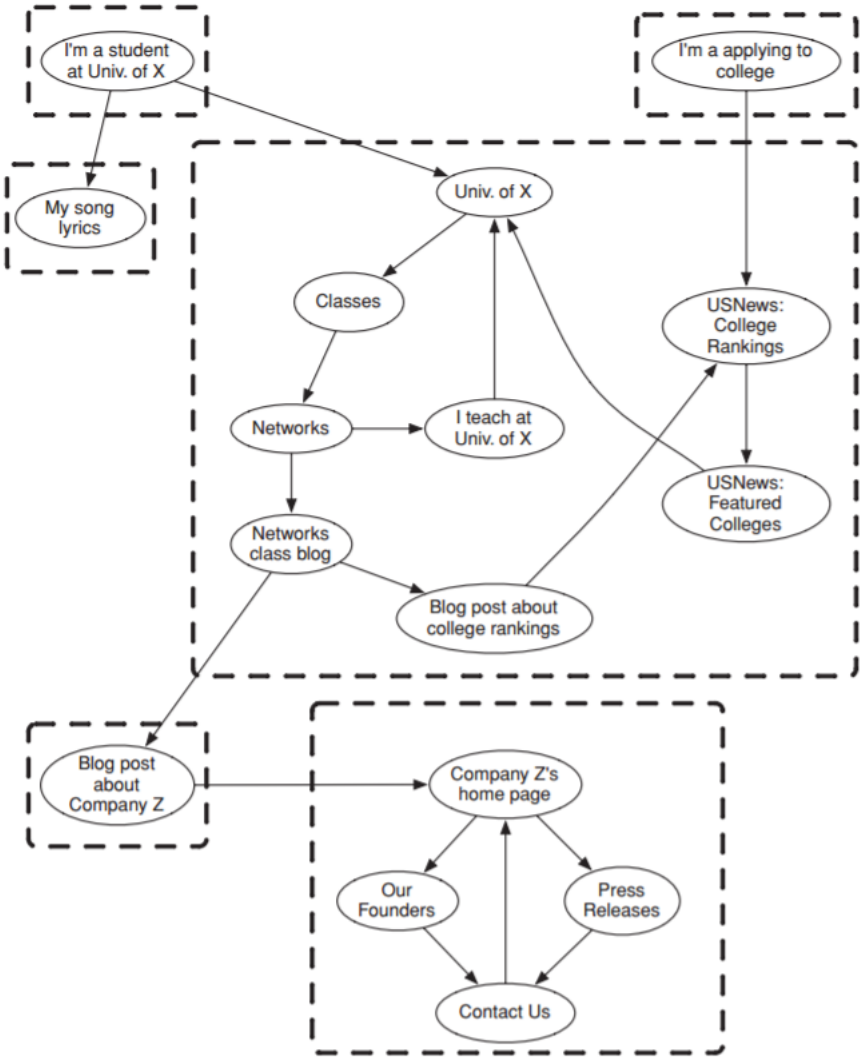


Figure 2: The directed graph in Figure 1 with its strongly connected components identified. (*Kleinberg 13.3*).

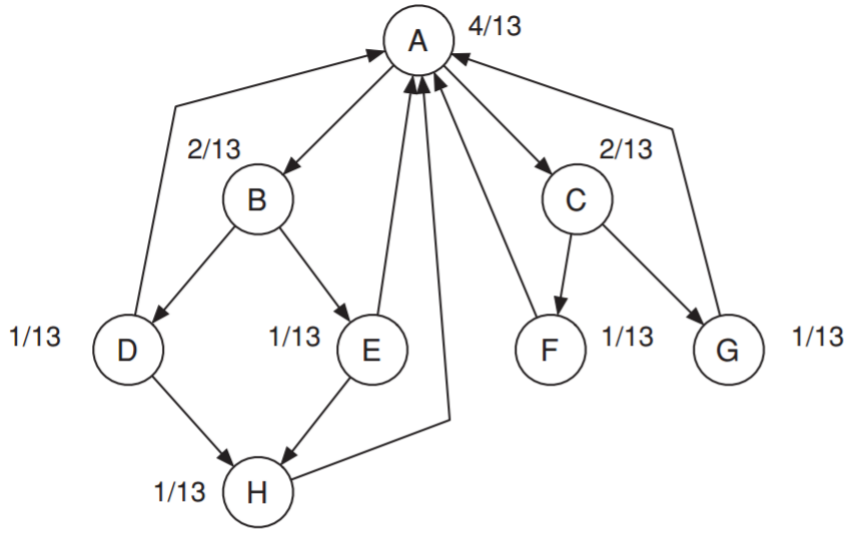


Figure 3: A collection of eight pages with their equilibrium PageRank values: page A has the largest PageRank, followed by pages B and C (which collect endorsements from A). (*Kleinberg 14.3*).

The intuition behind PageRank starts with simple voting based on in-links, and refines it using the principle of repeated improvement. In particular, the principle is applied here by having nodes repeatedly pass endorsements across their outgoing links, with the weight of a nodes endorsement based on the current estimate of its PageRank: nodes that are currently viewed as more important get to make stronger endorsements.

The basic algorithm of PageRank is as follows:

ALG 1

input: Graph G with M nodes, number of steps t

1. Assign all nodes the same initial PageRank value, $1/n$;
2. For t iterations:
   Update all PageRank values according to the *Basic PageRank Update Rule*;
   End for

return: The PageRank value of each node

Definition. ***Basic PageRank Update Rule****: Each page divides its current PageRank equally across its outgoing links and passes these equal shares to the pages it points to. (If a page has no outgoing links, it passes all its current PageRank to itself.) Each page updates its new PageRank to be the sum of the shares it receives.*

An example should help clarify how PageRank works. Figure 3 shows a graph on which we perform

the PageRank algorithm. The PageRank value of each node is as follows after each update:

$$\begin{bmatrix} k & A & B & C & D & E & F & G & H \\ 0 & 1/8 & 1/8 & 1/8 & 1/8 & 1/8 & 1/8 & 1/8 & 1/8 \\ 1 & 1/2 & 1/16 & 1/16 & 1/16 & 1/16 & 1/16 & 1/16 & 1/8 \\ 2 & 3/16 & 1/4 & 1/4 & 1/32 & 1/32 & 1/32 & 1/32 & 1/16 \end{bmatrix}$$

Note that the total PageRank in the network will remain constant as we apply these steps: since each page takes its PageRank, divides it up, and passes it along links, PageRank is neither created nor destroyed, just moved around from one node to another.

We can also think about PageRank in terms of transition matrices, which allow us to analyze how the PageRank values of a given graph might converge as the number of updates becomes large. Let M be an n by n matrix, where n is the number of nodes in the network and M_{ij} is the share of page i's PageRank that j should get in one update step. Thus, we have:

$$M_{ij} = \begin{cases} 1/d(i) & \text{if } i \text{ points to } j \\ 1 & \text{if } d(i) = 0 \text{ and } i = j \\ 0 & \text{otherwise} \end{cases}$$

where $d(i)$ is the outgoing degree of i. Figure 4 shows an example of a network and a corresponding matrix M. If we represent the PageRanks of all nodes at step t using a vector $r^{(t)}$, then to calculate an entry in $r^{(t)}$ we have:

$$r_j^{(t)} = M_{1j} \cdot r_1^{(t)} + \cdots + M_{nj} \cdot r_n^{(t)}.$$

This corresponds to multiplication by the transpose of our matrix M so we have that:

$$r^{(t)} = M^T \cdot r^{(t-1)}$$

In terms of our initial PageRank values at $t = 0$, then, we get:

$$r^{(t)} = (M^T)^t \begin{bmatrix} 1/n \\ \cdots \\ 1/n \end{bmatrix}.$$

This all looks a lot like what we did with the voter model and random walks last week. Intuitively, we realize that if we start at a random node in the graph, the probability of being at a page X after k steps of a random walk using our transition matrix M is precisely the PageRank of X after k applications of the Basic PageRank Update Rule! Earlier, we asked how, if at all, the PageRank values converge towards some equilibrium. In general, one can show that if the network is strongly connected, then there is a unique set of equilibrium values, and so whenever limiting PageRank values exist, they are the only values that satisfy this equilibrium. Figure 3 also shows the equilibrium PageRank values of each node.

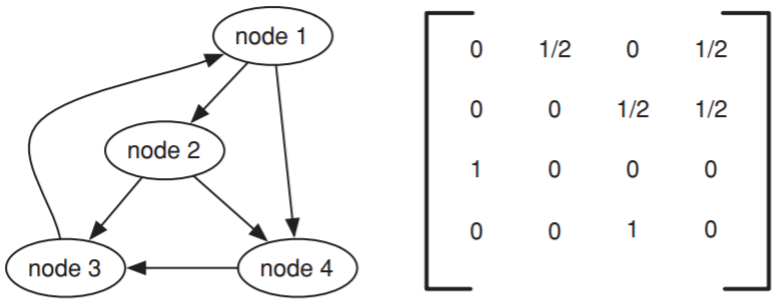


Figure 4: The flow of PageRank under the Basic PageRank Update Rule can be represented using a "transition" matrix M. (*Kleinberg 14.6*).

4 Scaled PageRank

There is one thing missing from our current definition of PageRank that is present in the actual definition of PageRank that is used in practice. Namely, consider Figure 5, a modification of Figure 3, only nodes F and G now point to each other instead of A. Over many updates, we would tend to observe that the PageRank in the network gradually flows into the "sink" SCC of F and G from the other nodes, such that eventually the only nodes in the network with non-zero PageRank are F and G. To combat this issue, we introduce a *scaling* factor s that should be strictly between 0 and 1 (typically around 0.8 or 0.9) and replace the Basic PageRank Update Rule with the following:

> **Definition.** ***Scaled PageRank Update Rule****: First apply the Basic PageRank Update Rule. Then scale down all PageRank values by a factor of s. We divide the residual $1 - s$ units of PageRank equally over all nodes, giving $\frac{1-s}{n}$ to each.*

In layman's terms, we can think of the Scaled PageRank Update Rule as introducing a "teleportation factor" to our algorithm, such that some small fraction of the PageRank of each page will "jump" from to every other page, even those that aren't connected by an edge. This helps us avoid the problem of a "sink" absorbing all of the PageRank in a network! If we are applying the Scaled PageRank Update Rule to our transition matrix M, we calculate new values in our new "scaled" matrix $\hat{M}$ as follows:

$$\hat{M}_{ij} = sM_{ij} + \frac{1-s}{n}$$

Then the following equations all hold true:

$$r_j^{(t)} = \hat{M}_{1j} \cdot r_1^{(t)} + \cdots + \hat{M}_{nj} \cdot r_n^{(t)}.$$

$$r^{(t)} = \hat{M}^T \cdot r^{(t-1)}$$

$$r^{(t)} = (\hat{M}^T)^t \begin{bmatrix} 1/n \\ \cdots \\ 1/n \end{bmatrix}.$$

Figure 6 demonstrates what such a matrix $\hat{N}$ would look like using our network from Figure 4. In a similar vein to our earlier analysis, if we start at a random node in the graph, the probability of

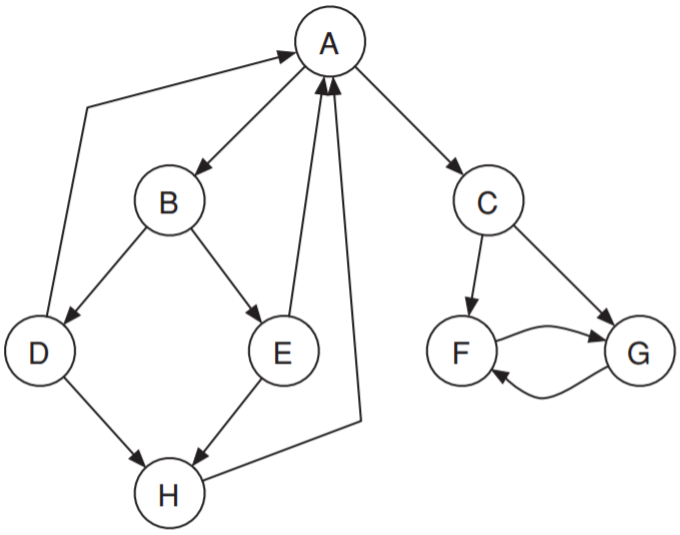


Figure 5: The same collection of eight pages, but F and G have changed their links to point to each other instead of to A. Without a "smoothing" effect, all the PageRank would go to F and G. (*Kleinberg 14.3*).

being at a page X after k steps of a random walk using our transition matrix $\hat{M}$ is precisely the PageRank of X after k applications of the Scaled PageRank Update Rule.

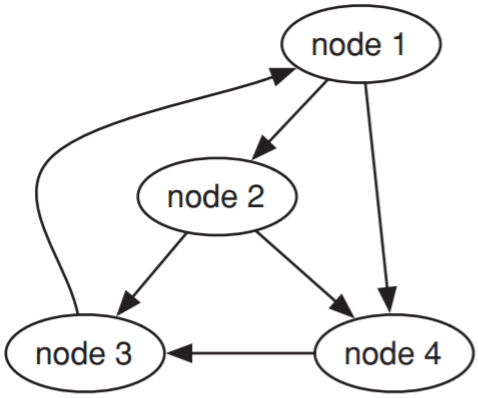
When do the PageRank values converge when we utilize the Scaled PageRank Update Rule? It turns out that they *always* do. To prove this, we have to brush up on our linear algebra and reference a powerful theorem called *Perron's Theorem.*

Definition. *(Review) For a square matrix M, a scalar λ is an* ***eigenvalue*** *if there exists a nontrivial solution x such that $Ax = \lambda x$. x is called the* ***eigenvector*** *corresponding to λ.*

Theorem. *(Perron's Theorem) For any matrix P whose entries are all positive:*

1. *P has a real eigenvalue $\lambda_1 > 0$ such that $\lambda_1 > |\lambda'|$ for all other eigenvalues λ'.*
2. *There is an eigenvector x with positive real terms corresponding to the largest eigenvalue λ_1, and x is unique up to multiplication by a constant.*
3. *If $\lambda_1 = 1$, then for any starting vector $y \neq \vec{0}$ with non-negative coordinates, where $\vec{0}$ is a vector whose terms all equal zero, the sequence of vectors $P^k y$ converges to a vector in the direction of x as k goes to infinity.*

If we interpret Perron's Theorem with our scaled transition matrix $\hat{M}$, we find that there is a unique vector x that remains fixed under the application of the scaled update rule, and that repeated application of the scaled update rule from any starting point converges to x. This vector



$$\begin{bmatrix} .05 & .45 & .05 & .45 \\ .05 & .05 & .45 & .45 \\ .85 & .05 & .05 & .05 \\ .05 & .05 & .85 & .05 \end{bmatrix}$$

Figure 6: A "transition" matrix $\hat{M}$ under the Scaled PageRank Update Rule (with scaling factor $s = 0.8$) and the corresponding network (*Kleinberg 14.6*).

x, the eigenvector corresponding to the largest real eigenvalue of $\hat{M}$ (which will always be 1 since each of the rows in $\hat{M}$ sums to 1), thus corresponds to the set of equilibrium PageRank values for Scaled PageRank.